Test Plan for Converting Digits

1. Introduction

1.1 Objectives

The objective of this test plan is to ensure the functionality and reliability of the Number to Words in web application. The primary focus is to test the conversion of numerical inputs into English words with “DOLLARS” and “CENTS” suffixes, and the correct storage and retrieval of these values from the In-Memory database.

1.2 Scope

- **Backend**
  - Convert Controller
  - Number Converter
  - Numbers model
- **Frontend**
  - Submission of numerical inputs
  - Display of converted values with “DOLLARS” and “CENTS” suffixes

2. Test Strategy

2.1 Types of Testing

- **Unit Testing**: Testing individual components
- **Integration Testing**: Testing the interaction between ConvertController and Number Converter, as well as between frontend and backend
- **Functional Testing**: Verifying the functionality against the requirements

2.2 Test Environment

- **Backend**: ASP.NET Core with a In-Memory Database
- **Frontend**: React with Vite
- **Tools**: Vitest with Testing Library (for frontend), xUnit (for backend unit tests)

3. Test Cases

3.1 Unit Test Cases for NumberConverter

Test Case ID	Description	Method Name	Input	Expected Output
UT-01	Test converting valid dollars and cents	ConvertingWords	Dollars="123", Cents="45"	Dollars="ONE HUNDRED AND TWENTY-THREE", Cents="FORTY-FIVE"
UT-02	Test converting valid dollars without cents	ConvertingWords	Dollars="123", Cents=""	Dollars="ONE HUNDRED AND TWENTY-THREE", Cents=""
UT-03	Test converting valid dollars only	ConvertIntegersToWords	123	"ONE HUNDRED AND TWENTY-THREE"
UT-04	Test converting valid cents only	ConvertFloatingPointToWords	45	"FORTY-FIVE"
UT-05	Test converting zero dollars	ConvertIntegersToWords	0	"ZERO"
UT-06	Test converting negative dollars	ConvertIntegersToWords	-123	"MINUS ONE HUNDRED AND TWENTY-THREE"
UT-07	Test converting tens and units	ConvertIntegersToWords	23	"TWENTY-THREE"
UT-08	Test converting numbers between 11 and 19	ConvertIntegersToWords	15	"FIFTEEN"
UT-09	Test converting hundreds	ConvertIntegersToWords	300	"THREE HUNDRED"
UT-10	Test converting thousands	ConvertIntegersToWords	4000	"FOUR THOUSAND"
UT-11	Test converting millions	ConvertIntegersToWords	5000000	"FIVE MILLION"

3.2 Integration Test Cases for ConvertController

Test Case ID	Description	Method Name	Input	Expected Status Code	Expected Output
IT-01	Test to get all numbers	GetAllNumbers	-	200 (OK)	-
IT-02	Test to get a specific number	GetNumber	1	200 (OK)	Dollars="ONE", Cents=""
IT-03	Test to post a valid number	PostNumber	Dollars="123", Cents="45"	201 (Created)	Dollars="ONE HUNDRED AND TWENTY-THREE", Cents="FORTY-FIVE"

IT-04	Test to post an invalid number	PostNumber	Dollars=null, Cents=null	400 (Bad Request)	-

3.3 Functional Test Cases for Frontend

Test Case ID	Description	Test Steps	Expected Output
TC-01	Renders the heading and form	1. Render the App component.	- The heading “Conversion Numbers” is displayed. - The input field with the label “Type a number” is displayed. - The “Convert” button is displayed.
TC-02	Shows error when input is not a number	1. Render the App component. 2. Enter “abc” in the input field. 3. Observe the displayed error message.	- The error message “Please enter a valid number.” is displayed.
TC-03	Shows error when number is greater than a billion	1. Render the App component. 2. Enter “1000000000” in the input field. 3. Click the “Convert” button. 4. Observe the displayed error message.	- The error message “Please enter a valid number less than a billion. or with two decimal places.” is displayed.
TC-04	Handles valid form submission	1. Mock a successful fetch response with the data { id: 1, dollars: 'ONE HUNDRED', cents: '' }. 2. Render the App component. 3. Enter “100” in the input field. 4. Click the “Convert” button. 5. Wait for the converted result to be displayed.	- The error messages are not displayed. - The text “ONE HUNDRED DOLLARS” is displayed.
TC-05	Handles fetch error	1. Mock a fetch rejection with an error “API is down”. 2. Render the App component. 3. Enter “100.00” in the input field. 4. Click the “Convert” button. 5. Wait for the error message to be displayed.	- The error message “An error occurred while converting the number.” is displayed.